## Python: module regrid.pressure

# regrid.pressure

index

## **Modules**

MA regrid regrid string

Numeric copy

### Classes

#### **PressureRegridder**

```
class PressureRegridder

#-----

#

# PURPOSE: To perform all the tasks required to regrid the input the pressure dimension only.

# PROCEDURE: Step One:

# Make an instance of class PressureRegridder pass Step Two:

# Pass the input data with some descriptive parame in return
```

#### Methods defined here:

```
__call__(self, ar, missing=None, order=None, method='log')
```

Call the pressure regridder function.

ar is the input array, a variable, masked array, or Numeric a missing is the missing data value, if any. It defaults to the defined for the input array, if any.

order is of the form "tzyx", "tyx", etc.

method is either 'log' to interpolate in the log of pressure,

#### \_\_init\_\_(self, axisIn, axisOut)

#
# PURPOSE: To make an instance which entails setting up th

```
def __init__(self, levIn, levOut):
                 #
                               PROCEDURE:
                 #
                                            The user must assemble two pieces of information:
                 #
                                                        axisIn - the input level axis
                                                        axisOut - the output level axis
                            USAGE:
                                                  To make an instance preparing for a regrid along
                                                           r = <u>PressureRegridder</u>(levIn, levOut)
rgrd(self, dataIn, missingValueIn, missingMatch, logYes='yes', positionIn=None, missingValueOu
                            PURPOSE: To perform all the tasks required to regrid the
                 #
                                                            dataout along the level dimension only.
                 #
                            DEFINITION:
                                                            def rgrd(self, dataIn, missingValueIn, missingNalueIn, mi
                            PASSED: dataIn -- data to regrid
                 #
                                                           missingValueIn -- the missing data value to use
                 #
                                                                                                                   and there are two choices:
                                                                                                                                     None -- there is no mis
                 #
                                                                                                                                     A number -- the value t
                                                                                                                   The presence of missing data
                 #
                 #
                                                           missingMatch -- the comparison scheme used in s
                                                                                                             in as missingValueIn. The choice
                                                                                                                            None -- used if None is th
                                                                                                                            exact -- used if missingVa
                                                                                                                            greater -- the missing dat
                 #
                                                                                                                            less -- the missing data v
                 #
                                                            logYes -- choose the level regrid as linear in
                                                                                           'yes' for log. Anything else is linea
                                                            positionIn -- a tuple with the numerical positi
```

DEFINITION:

```
#
                             in C or Python order specified in
                             latitude, level and time. Longitu
                             required. If time is missing subm
                             tuple. Notice that the length of
                             Explicitly, in terms of the shape
                                  positionIn[0] contains the p
                                  positionIn[1] contains the p
                                  positionIn[2] contains the p
                                  positionIn[3] contains the p
                             As examples:
                                  If the C order shape of 4D o
                                      (number of longitudes, r
                                  submit
                                       (0, 3, 2, 1)
#
                                  If the C order shape of 3D o
                                      (number of longitudes, r
                                  submit
                                      (0, 2, 1, None)
                             Send in None if the shape is a su
                             latitude, longitude) which is eva
                                3D -- code assumes (2,1,0,None)
                                4D -- code assumes (3, 2, 1, 0)
               missingValueOut -- the value for the missing of
                                   default entry, None, the co
                                   1.0e20
    RETURNED : dataOut -- the regridded data
#
    USAGE:
           Example 1. To regrid dataIn into dataOut using al
#
                       missing data.
               dataOut = x.rgrd(dataIn, None, None)
           Example 2. To regrid dataIn into dataOut using 1.
                       dataOut = x.rgrd(dataIn, 1.e20, 'great
```

## **Functions**

```
checkorder(positionIn)
            -----
    #
        purpose: construct the tuples for transposing the data to st
                inverse for transposing it back to the original dim
    #
        usage: newOrder, inverseOrder = checkorder(positionIn)
    #
      passed: positionIn -- array with location of longitude, lat
                           in the sense of the python shape of t
    #
       returned: newOrder -- tuple to transpose data to the order (
                inverseOrder -- tuple to transpose data to back to
    #-----
sendmsg(msg, value1=None, value2=None)
    #
       purpose: send the same message to the screen
    # passed : msg - the string
               value - the number associated with the string
       returned: return
    #------
```